Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра алгоритмических языков

Гулак Максим Андреевич

Проверка унимодулярности матрицы дифференциальных операторов

КУРСОВАЯ РАБОТА

Научный руководитель:
к.ф.-м.н. А.А. Панфёров

Москва, 2023

Содержание

1 Введение **3**

2 Постановка задач **5**

3 Система компьютерной алгебры SageMath **6**

3.1 Обзор системы SageMath 6

3.2 Особенности системы SageMath 6

3.3 Работа с SageMath 7

4 Матрицы дифференциальных операторов **8**

4.1 Основные понятия 8

4.2 Элементарные операции над строками 9

4.3 Приведение матрицы по строкам 9

4.4 Алгоритм Row-Reduction 11

4.5 Построение обратной матрицы 12

5 Многочлены Оре **14**

5.1 Определение 14

5.2 Поддержка в SageMath 15

6 Реализация **16**

6.1 Описание алгоритма 16

6.2 Примеры работы 16

7 Заключение **18**

Литература **19**

1 Введение

Матричное исчисление имеет широкое применение в различных отраслях науки. Проверка того, обратима ли данная матрица над полем или кольцом, и вычисление обратной матрицы являются классическими математическими задачами.

В работе рассматриваются матрицы, элементы которых принадлежат кольцу скалярных линейных дифференциальных операторов над дифференциальным полем, т.е. над полем, снабжённым операцией дифференцирования. В этом случае вместо понятия обратимости матрицы используется понятие *унимодулярности* матрицы, которое и будет использовано далее.

Такие матрицы возникают при работе с линейными дифференциальными системами, которые в свою очередь возникают во многих приложениях, таких как системы многих тел, модели электрических цепей, моделирование роботов, механические системы и т.д.

Для решения задачи проверки унимодулярности операторной матрицы и построения обратной матрицы могут использоваться некоторые известные алгоритмы преобразования матриц с помощью обратимых операций по строкам или столбцам. Такими являются, например, алгоритмы построения форм Эрмита или Джекобсона, а также алгоритм Row-Reduction.

Для работы с операторными матрицами, как правило, используются специализированные системы компьютерной алгебры. Такой системой называется программное обеспечение, предназанченное для выполнения математических операций и алгебраических вычислений с символьными выражениями вместо числовых значений.

На данный момент существует большое количество систем компьютерной алгебры. Некоторые из них являются платными, например, Maple и Mathematica, а некоторые, такие как Maxima, SageMath или библиотека SymPy для языка Python, распространяются свободно. В работе используется система SageMath по причине своей доступности и идентичного языку Python синтаксису.

2 Постановка задач

В курсовой работе требовалось:

1. Изучить систему компьютерной алгебры SageMath.

2. Изучить существующие алгоритмы проверки унимодулярности.

3. Реализовать алгоритм проверки унимодулярности матрицы дифференциальных операторов в системе компютерной алгебры SageMath.

3 Система компьютерной алгебры SageMath

3.1 Обзор системы SageMath

SageMath — это свободная и открытая система компьютерной алгебры, разработанная для решения широкого спектра математических задач. Она объединяет в себе множество различных математических пакетов и инструментов, обеспечивая пользователю возможность работы в единой среде.

Цель создания системы SageMath заключалась в предоставлении математикам, ученым и инженерам простого и эффективного способа решения математических задач, без необходимости использования различных инструментов и пакетов. Она позволяет проводить множество операций, начиная от простых математических вычислений, заканчивая решением сложных дифференциальных уравнений и построением трехмерных графиков.

Одной из главных особенностей системы SageMath является её открытый и расширяемый код, что позволяет пользователям модифицировать и дополнять функциональность системы в соответствии с их потребностями. SageMath также поддерживает различные языки программирования, такие как Python, Cython, C, Fortran и др., что позволяет пользователям писать свои собственные модули и расширения.

3.2 Особенности системы SageMath

SageMath предоставляет по сравнению с другими системами компьютерной алгебры преимущества, в числе которых:

1. Бесплатность и открытый код — SageMath предоставляет пользователю возможность использовать систему бесплатно и иметь доступ к её исходному коду.

2. Широкий спектр функциональности.

3. Кроссплатформенность — SageMath может работать на различных операционных системах, включая Windows, Mac OS и многих дистрибутивах Linux.

4. Простота в использовании — система предоставляет простой и понятный интерфейс пользователя, что делает её использование доступным даже для начинающих пользователей. Также, поскольку синтаксис в SageMath идентичен синтаксису Python, освоение этой системы не требует дополнительного изучения принципов написания программ в ней.

3.3 Работа с SageMath

Существует несколько вариантов работы в SageMath: терминал, веб-интерфейс SageMathCell и Jupyter Notebook.

Первый вариант похож на обычный интерпретатор Python, который удобно использовать для небольших вычислений, поэтому для реализаций больших программ он не подходит.

Во втором варианте уже больше возможностей для написания большого количества кода. Однако работа в SageMathCell требует подключения к интернету, а также не позволяет разбивать код на несколько частей (ячеек), из-за чего при небольшом изменении в программе приходится запускать все команды заново.

Последний способ работы с SageMath исправляет многие недостатки предыдущих. Работа в формате ноутбука (.ipynb) позволяет разделять код на ячейки, выполянять их в любом порядке и при желании смотреть промежуточные результаты их выполнения.

4 Матрицы дифференциальных операторов

4.1 Основные понятия

Пусть K — дифференциальное поле характеристики 0 с производной $\partial = \ '$. В дальнейшем будут использоваться следующие обозначения. Кольцо дифференциальных операторов с коэффициентами в K обозначается через $K[\partial]$. Кольцо матриц размера $m \times m$ с элементами, принадлежащими кольцу R обозначается через $\mathrm{Mat}_m(R)$.

Рассмотрим систему линейных дифференциальных уравнений

$$L \cdot y(x) = 0,$$

где $L \in \mathrm{Mat}_m(K[\partial])$, $y(x)$ — m–мерный вектор с компонентами $y_1(x), ..., y_m(x)$.

L является матрицей дифференциальных операторов и может быть записана в виде

$$L = A_r \partial^r + A_{r-1} \partial^{r-1} + \cdots + A_0,$$

где $A_i \in \mathrm{Mat}_m(K), i = 0, ..., r$.

Матрица A_r ненулевая и называется *ведущей* матрицей L. Число r называется порядком L и обозначается как $\mathrm{ord}\, L$.

Для обозначения i-й строки L используется запись $L_{i,*}$.

Строки $L_{1,*}, ..., L_{s,*} (s \leq m)$ называются независимыми над $K[\partial]$, если из того, что $f_1 L_{1,*} + \cdots + f_s L_{s,*} = 0$ $(f_1, \ldots, f_s \in K[\partial])$ следует, что $f_1 = \ldots = f_s = 0$. *Ранг* L — максимальное количество независимых строк.

Пусть δ_i — порядок строки $L_{i,*}$. Вектор $\delta = (\delta_1, ..., \delta_m)$ называется вектором порядков строк L. *Фронтальной матрицей* называется такая матрица L_0, что $(L_0)_{i,*} = (A_{\delta_i})_{i,*}$.

Матрица дифференциальных операторов $U \in \mathrm{Mat}_m(K[\partial])$ называется *унимодулярной* (или *обратимой*), если существует $U^{-1} \in \mathrm{Mat}_m(K[\partial])$ такая, что $U^{-1}U = UU^{-1} = I_m$, где I_m — единичная матрица размера $m \times m$.

4.2 Элементарные операции над строками

К элементарным операциям над строкам относятся следующие:

1. Перестановка двух строк.

2. Умножение строки слева или справа на ненулевой элемент из K.

3. Сложение строки с другой строкой, умноженной слева на скалярный дифференциальный оператор из $K[\partial]$.

Каждая элементарная операция по строкам соответствует умножению на элементарную матрицу.

4.3 Приведение матрицы по строкам

Матрица L называется *приведённой по строкам*, если ненулевые строки её фронтальной матрицы независимы над $K[\partial]$. В случае, если L полного ранга, это означает обратимость фронтальной матрицы.

Следующие три леммы показывают, что любая матрица дифференциальных операторов может быть преобразована к приведённому по строкам виду с помощью элементарных операций над строками:

Лемма 1 ([1], [2]). *Пусть $L \in \mathrm{Mat}_m(K[\partial])$ и $U, V \in \mathrm{Mat}_m(K[\partial])$ являются унимодулярными. Тогда ранги L, UL и LV равны.*

Лемма 2 ([1], [2]). *Ранг приведённой по строкам матрицы дифференциальных операторов равен рангу её фронтальной матрицы.*

Лемма 3 ([1], [2]). *Пусть* $L \in \mathrm{Mat}_m(K[\partial])$ *ранга* $s \le m$. *Тогда всегда можно построить унимодулярную матрицу* $U \in \mathrm{Mat}_m(K[\partial])$ *такую, что:*

$$UL = \left[\begin{array}{c} L^* \\ \hline 0 \end{array}\right],$$

где L^* *— приведённая по строкам матрица дифференциальных операторов размера* $s \times m$ *такая, что* $\mathrm{ord}\ L^* \le \mathrm{ord}\ L$ *и все её строки ненулевые.*

Для полноты изложения приведём доказательство Леммы 3.

Доказательство. Если L уже приведена по строкам, то $U = I_m$, и дальнейшие вычисления проводить не нужно. Иначе, мы можем предположить, без потери общности, что L имеет все свои нулевые строки в нижней части матрицы. В таком случае фронтальная матрица L имеет вид:

$$\left[\begin{array}{c} L_0 \\ \hline 0 \end{array}\right],$$

где L_0 — фронтальная матрица первых k строк L ($k \ge s$). Так как L не приведена по строкам, то L_0 имеет ранг меньший k. Тогда мы всегда можем найти ненулевой вектор $v = (v_1, \dots, v_k)$, такой, что $vL_0 = 0$. Выберем индекс t, такой, что $v_t \ne 0$ и $\delta_t = \max(\delta_i \ne 0)$. Определим $U_1 = \mathrm{diag}(U_{11}, I_{m-k})$, где

$$U_{11} = \begin{bmatrix} 1 &&&&&&& \\ & \ddots &&&&&& \\ && 1 &&&&& \\ v_1\partial^{\delta_t-\delta_1} & \cdots & v_{t-1}\partial^{\delta_t-\delta_{t-1}} & v_t & v_{t+1}\partial^{\delta_t-\delta_{t+1}} & \cdots & v_k\delta^{\delta_t-\delta_k} \\ &&&& 1 && \\ &&&&& \ddots & \\ &&&&&& 1 \end{bmatrix}.$$

Тогда U_1 является унимодулярной и при умножении на L слева оставляет без изменений строки с индексами $i \neq t$ и заменяет t-ю строку на

$$\sum_{i=1}^{m} v_i \partial^{\delta_t' - \delta_i'} L_{i,*}' = \sum_{i=1}^{m} v_i \ell c(L_{i,*}) \partial^{\delta_t} + p = v L_0 \partial^{\delta_t} + p = 0 + p = p,$$

где p обозначает элементы порядка, меньшего δ_t.

Таким образом, t-я строка $U_1 L$ обладает порядком, меньшиим, чем t-я строка L. Повторяя этот процесс конечное число раз, мы получим унимодулярную матрицу дифференциальных операторов U, такую, что UL будет иметь вид

$$UL = \left[\frac{L^*}{0} \right],$$

где L^* — приведённая по строкам матрица дифференциальных операторов размера $m_1 \times m$ ($m_1 \leq k$) с ненулевыми строками. Остаётся показать, что $m_1 = s$. Согласно лемме 1, $\operatorname{rank} L = \operatorname{rank} UL = \operatorname{rank} L^*$. С одной стороны, согласно лемме 2, ранг L^* равен рангу её фронтальной матрицы, то есть m_1. С другой стороны, $\operatorname{rank} L = s$, поэтому $m_1 = s$. □

4.4 Алгоритм Row-Reduction

Из доказательства леммы 3 следует алгоритм приведения матрицы по строкам, получивший название Row-Reduction. Дополнительно при этом строится унимодулярная матрица, которая пригодится далее для построения обратной матрицы.

На вход алгоритма поступает матрица дифференциальных операторов $L \in \mathrm{Mat}_m(K[\partial])$ с порядком по строкам $\delta = (\delta_1, ..., \delta_m)$ и порядком $\operatorname{ord} \ L = l$.

Далее необходимо инициализировать матрицу $L^{'} = L$ с порядком по строкам $\delta^{'} = (\delta^{'}_1, ..., \delta^{'}_m)$, соответствующую ей фронтальную матрицу $L^{'}_0$ и матрицу $U = I_m$.

Пока ненулевые строки $L^{'}_0$ зависимы над необходимо выполнять следующие шаги:

1. Вычислить ненулевой вектор $v = (v_1, ..., v_m) \in K^{1\times m}$ такой, что $vL^{'}_0 = 0$.

2. Выбрать целое число t такое, что $v_t \neq 0, \delta^{'}_t = \max(\delta^{'}_i, v_i \neq 0)$.

3. Заменить $L^{'}_{t,*}$ на $\sum\limits_{i=1}^{m} v_i \partial^{\delta^{'}_t - \delta^{'}_i} L^{'}_{i,*}$.

4. Заменить $U_{t,*}$ на $\sum\limits_{i=1}^{m} v_i \partial^{\delta^{'}_t - \delta^{'}_i} U_{i,*}$.

5. Обновить $L^{'}_0$ и $\delta^{'}$.

В результате на месте $L^{'}$ получается приведённая матрица дифференциальных операторов и унимодулярная матрица дифференциальных операторов U такая, что $L^{'} = UL$.

4.5 Построение обратной матрицы

В статье [3] приводится следующий факт, применимый к матрицам дифференциальных операторов полного ранга:

Утверждение 1. *Пусть $L \in \mathrm{Mat}_m(K[\partial])$ имеет обратимую фронтальную матрицу. Тогда L унимодулярна тогда и только тогда, когда* $\mathrm{ord}\ L = 0$.

Алгоритм Row-Reduction, описанный в разделе 4.4, позволяет вычислить унимодулярную матрицу $U \in \mathrm{Mat}_m(K[\partial])$, такую, что $L^{'} = UL$ имеет обратимую фронтальную матрицу. Утверждение 1 означает, что L унимодулярна тогда и только тогда, когда $L^{'}$ — обратиимая матрица в $\mathrm{Mat}_m(K)$. В таком случае $(L^{'})^{-1}UL = I_m$. Таким образом,

$$L^{-1} = (L^{'})^{-1}U. \tag{1}$$

5 Многочлены Оре

Теория колец многочленов Оре даёт возможность рассматривать различные операторы, в частности линейные дифференциальные, с общей точки зрения. Это позволяет создавать многоцелевые алгоритмы и соответствующие программы, которые можно настраивать на конкретный вид операторов и уравнений.

Поэтому для представления операторных матриц в системе компьютерной алгебры SageMath удобно использовать многочлены Оре.

5.1 Определение

Пусть k — поле характеристики 0, $\sigma : k \to k$ — автоморфизм k. Дифференцирование относительно σ — это любое отображение $\delta : k \to k$, для которого $\forall a, b \in k$

$$\delta(a + b) = \delta a + \delta b$$

$$\delta(ab) = \sigma(a)\delta b + \delta a b.$$

Кольцо Оре (или *кольцо многочленов Оре*) над k, заданное посредством σ и δ и обозначаемое $k[x; \sigma; \delta]$, — это кольцо многочленов от x над k с обычным сложением многочленов и умножением, заданным формулой

$$xa = \sigma(a)x + \delta a, \ \forall a \in k.$$

В [5] показано, что при $k = \mathbb{F}(t)$ (где $\mathbb{F}$ — любое подполе $\mathbb{C}$) в дифференциальном случае в качестве σ используется тождественное отображение на k, а в качестве δ — $\frac{d}{dt}$.

5.2 Поддержка в SageMath

Для работы с многочленами Оре в SageMath существует встроенный функционал ([7]), который позволяет создавать кольца Оре над коммутативными кольцами.

К примеру, кольцо Оре дифференциальных операторов над $\mathbb{Q}[x]$ может определяться следующим образом:

```
R.<x> = QQ[]
der = R.derivation()
A = OrePolynomialRing(R, der, 'D')
```

Иной способ определения:

```
A.<d> = R['D', der]
```

В результате создаётся следующий объект:

```
In [1]: R.<x> = QQ[]
        der = R.derivation()
        A.<D> = R['D',der]

In [2]: A

Out[2]: Ore Polynomial Ring in D over Univariate Polynomial Ring in x over Rational Field twisted by d/dx
```

Примеры вычисления произведения в описанном кольце Оре:

```
In [4]: D*x

Out[4]: x*D + 1

In [5]: D*x^2

Out[5]: x^2*D + 2*x
```

6 Реализация

6.1 Описание алгоритма

Согласно выводам из раздела 4.5, можно сформулировать итоговый алгоритм проверки унимодулярности матрицы дифференциальных операторов.

1. Преобразовать исходную матрицу к приведённому по строкам виду.

2. Проверить, является ли полученная матрица обратимой в $\mathrm{Mat}_m(\mathbb{Q}(x))$.

3. Если является, то исходная матрица является унимодулярной, если нет, то не является.

Далее, в случае положительного ответа, можно построить обратную матрицу для исходной по формуле (1).

6.2 Примеры работы

Ниже рассмотерены примеры работы алгоритма.

```
sage: X
[     1/2*x^2 -1/2*x*D + 1]
[    -x*D - 3          D^2]
sage: constructInverse(X)
(
      [    D^2 1/2*x*D]
True, [x*D + 1 1/2*x^2]
)
sage: _[1]*X
[1 0]
[0 1]
```

```
sage: Y
[     D^2 1/2*x*D]
[x*D + 1 1/2*x^2]
sage: constructInverse(Y)
(
      [      1/2*x^2 -1/2*x*D + 1]
True, [     -x*D - 3          D^2]
)
sage: _[1]*Y
[1 0]
[0 1]
```

```
sage: Z
[              1/12 D + 1/3*x^2 + 2/3]
[               1/4      3*D + x^2 + 1]
sage: constructInverse(Z)
(
      [-36*D - 12*x^2 - 12     12*D + 4*x^2 + 8]
True, [                  3                   -1]
)
sage: _[1]*Z
[1 0]
[0 1]
```

```
sage: L
[   D^3 + x      2*D^2     x^2 + x]
[       D^2      x*D^2 2*x^2 + 1]
[         D        x*D           1]
sage: constructInverse(L)
(False, [])
```

7 Заключение

В ходе курсовой работы были получены следующие результаты:

1. Изучена система компьютерной алгебры SageMath.
2. Изучен алгоритм Row-Reduction для приведения операторных матриц по строкам.
3. Реализован алгоритм проверки унимодулярности матрицы дифференциальных операторов и построения обратной матрицы в системе компютерной алгебры SageMath.

Исходный код программы доступен по адресу: `https://github.com/MaxudMSU/unimodularity-course-work`.

Литература

[1] Beckermann B., Cheng H. and Labahn G. Fraction-free row reduction of matrices Ore polynomials. *Journal of Symbolic Computation*, 41(5):513–543, 2006.

[2] Barkatou M. A., El Bacha C., Labahn G., and Pflügel E. On simultaneous row and column reduction of higher-order linear differential systems. *Journal of Symbolic Computation*, 49(1):45–64, 2013.

[3] Abramov S. A. and Barkatou M. A. On solution spaces of products of linear differential or difference operators. *ACM Communications in Computer Algebra*, 48(4):155–165, 2014.

[4] Ore O. Theory of non-commutative polynomials. Annals of Mathematics, 34:480–508.

[5] Абрамов С. А., Ле Х. К. и Ли З. Кольца полиномов Оре одной переменной в компьютерной алгебре. *Современная математика и приложения*, 13:24–39, 2004.

[6] Kosan T. SAGE For Newbies. 2008.

[7] Sage Reference Manual. `https://doc.sagemath.org/html/en/reference` Дата обращения: 25 апреля 2023.